



Universidad Simón Bolívar

Departamento de Computación y Tecnología de la Información

Laboratorio de Algoritmos y Estructuras III CI-2693

Prof. Fernando Torre Mora

Especificaciones del Proyecto

Implementación del TAD Grafo Genérico Para Multígrafos

Integrantes:

José Barrera. Carnet: 15 - 10123

Jean Yazbek. Carnet: 15 – 11550

Sartenejas, octubre de 2018

Con la finalidad de implementar un TAD grafo genérico para multígrafos ajustado a las especificaciones del proyecto se ha estructurado el código en 12 archivos .java, a saber: Grafo, Arista, Arco, Lado, Vertice, Grafo Dirigido, Grafo No Dirigido, Cliente, Prueba, Transformer, TransformarBoolean, TransformarDouble, TransformarString. A continuación, se explicarán las decisiones de diseño y se indicarán los detalles más relevantes de la implementación realizada, con la finalidad de brindar al lector una idea general acerca de cómo funciona todo en conjunto.

En principio, se seguirá el orden en el que aparecen las implementaciones en el enunciado del proyecto:

- **Vertice<E>**: esta clase pública posee un constructor que juega el papel de la función solicitada Crear Vertice, y las funciones públicas getPeso, getId, getDato, toString, las cuales funcionan de manera trivial.
- **Lado<E>**: esta clase pública y abstracta posee las funciones públicas getPeso, getId, getDato, toString, las cuales no contienen ninguna instrucción.
- **Arco<E>**: esta clase pública extiende de la clase abstracta Lado, posee un constructor que juega el papel de la función solicitada Crear Arco, y las funciones públicas getExtremoInicial, getExtremoFinal, getPeso, getId, getDato, toString, las cuales funcionan de manera trivial.
- **Arista<E>**: análoga a la clase Arco<E>, se sustituye el nombre de las funciones getExtremoInicial, getExtremoFinal, por getExtremo1 y getExtremo2 para denotar que al tratarse de un grafo no dirigido en los lados no existe un orden.
- **Grafo<V,L>**: esta interfaz pública contiene todas las funciones indicadas, a saber: cargarGrafo, numeroDeVertices, numeroDeLados, agregarVertice, agregarVertice, obtenerVertice, estaVertice, estaLado, eliminarVertice, vertices, lados, grado, adyacentes, incidentes, clone, toString.
- **GrafoDirigido<V,L>**: esta clase pública es un implements de la interfaz Grafo, y posee las siguientes operaciones particulares: agregarArco, agregarArco, eliminarArco, obtenerArco, gradoInterior, gradoExterior, sucesores, predecesores. La idea general es que el digrafo es un diccionario (utilizamos el diccionario de Java: **Hashtable**) donde las claves son objetos tipo vertice, y los valores son listas

de objetos tipo lados. Es decir, cada vértice tiene asociado una lista de arcos donde dicho vértice es el extremo inicial.

- **GrafoNoDirigido<V,L>**: esta clase pública es un implements de la interfaz Grafo, y posee las siguientes operaciones particulares: agregarArista, eliminarArista, obtenerArista. La idea general es análoga a la de dígrafo, pero cada Arista está presente en la lista de lados de ambos vertices. Es decir, cada vértice tiene asociado una lista de arcos donde dicho vértice es el extremo 1 o el extremo 2.
- **Cliente**: esta clase pública es la que le permite al usuario interactuar con los métodos del TAD. En ella se encuentran cuatro funciones: terminar, menuDirigido, menuNoDirigido y main.
- **Prueba**: archivo que contiene a la clase pública Prueba, su propósito es crear un grafo no dirigido y un dígrafo y llama a los métodos de cada uno para mostrar su funcionamiento.
- **Transformer**: esta interfaz pública de la que luego hacen implements TransformarBoolean, TransformarDouble y TransformarString, sirve para pasar datos de un tipo a otro. Las tres anteriores transforman Strings en aquello que indica su nombre.

Ahora bien, respondiendo a la pregunta ¿Cómo Ejecutar El Cliente?, el usuario luego de introducir: java Cliente, puede colocar seguidamente un nombre de archivo en caso de que desee cargar un grafo, o no colocarlo, en cuyo el programa asume que se desea crear un nuevo grafo. Luego se pedirá al usuario que indique el tipo de dato para los vertices, luego el tipo de lado, y finalmente el tipo de grafo. Una vez hecho esto dependiendo del tipo de grafo seleccionado se mostrará o bien el menuDirigido o el menuNoDirigido ambos con la misma estructura, pero no con las mismas acciones disponibles. Después de seleccionar cada acción se volverá a desplegar el menú. El usuario puede usar la opción salir para cerrar el programa.